Implied Volatilty and Value-at-Risk Modeling

Kevin Sheppard

Today

- Implied Variance
- Value-at-Risk
    - Key concepts
    - Models
        - Volatility-based
        - Quantile Regression
        - Historical Simulation
    - Model evaluation

Implied Variance

- Implied volatility is very different from ARCH and Realized measures
- Market based: Level of volatility is calculated from options prices
- Forward looking: Options depend on future price path
- “Classic” implied relies on the Black-Scholes pricing formula
- “Model free” implied volatility exploits a relationship between the second derivative of the call price with respect to the strike and the risk neutral measure
- VIX is a Chicago Board Options Exchange (CBOE) index based on a model free measure
- Allows volatility to be directly traded

Black-Scholes Implied Volatility

- Prices follow a geometric Brownian Motion

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t$$

- Constant drift and volatility
- Price of a call is

$$C(T,K) = S\Phi(d_1) + Ke^{-rT}\Phi(d_2)$$

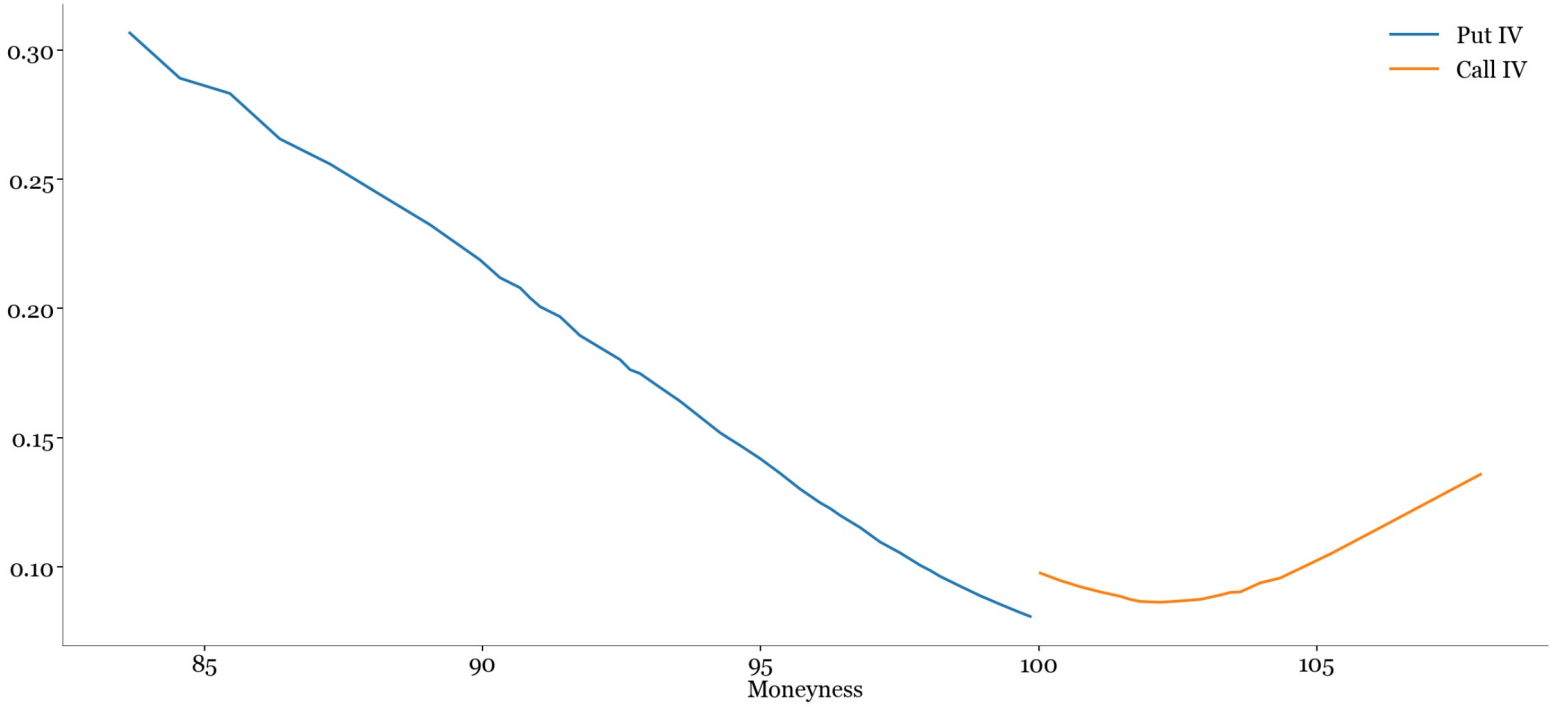
$$d_1 = \frac{\ln(S/K) + \left(r + \sigma^2/2\right)T}{\sigma\sqrt{T}},\ d_2 = \frac{\ln(S/K) + \left(r - \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

- Invert to produce a formula for the volatility given the call price $C(T,K)$

$$\sigma_t^{Implied} = g\left(C_t(T,K), S_t, K, T, r\right)$$

Black-Scholes Smile

In [2]:
```
bsiv()
```



Model-free Implied Variance

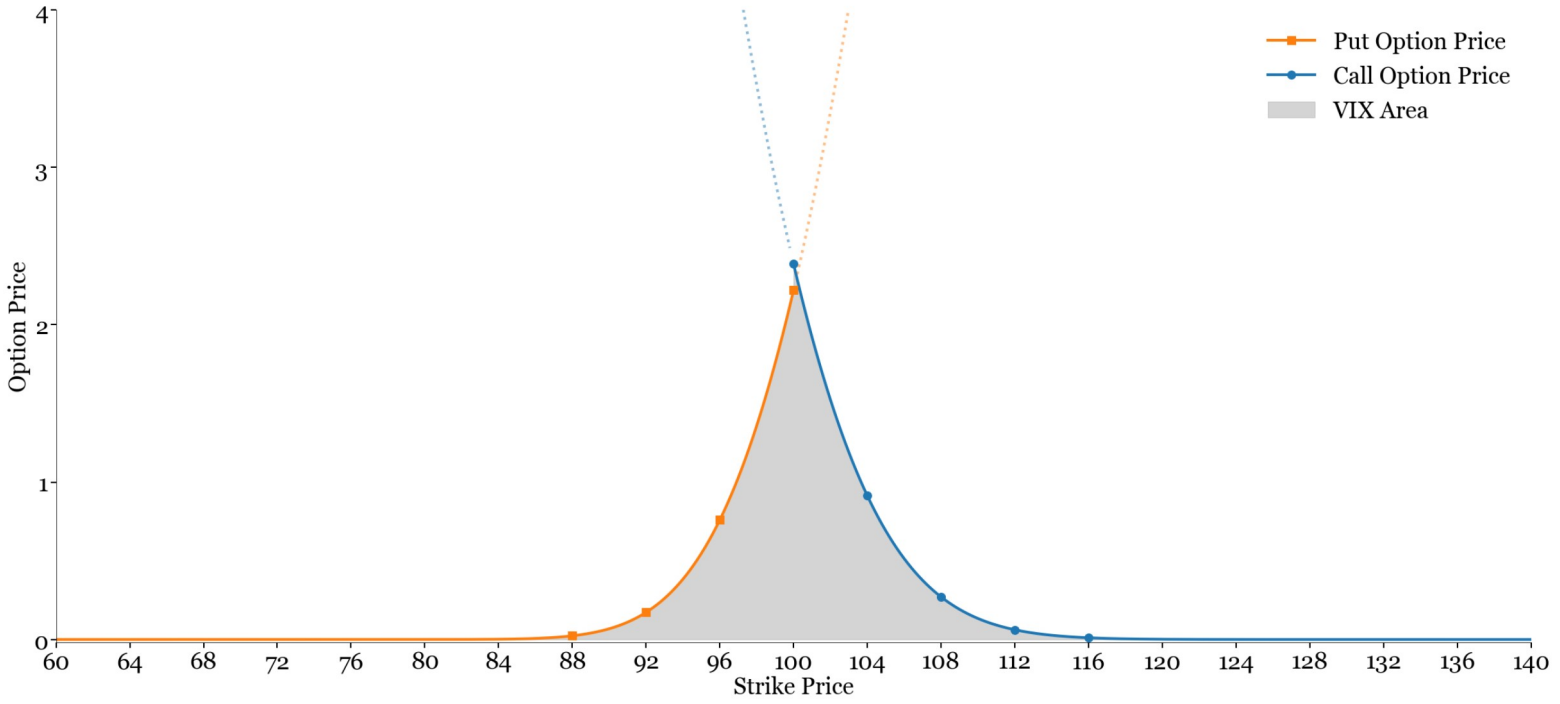
- VIX is continuously computed by the CBOE
- Uses both out-of-the-monry calls and puts

$$\sigma^2 = \frac{2}{T} e^{rT} \sum_{i=1}^{N} \frac{\Delta K_i}{K_i} \frac{Q(K_i)}{K_i} - \frac{1}{T} \left(\frac{F_0}{K_0} - 1 \right)^2$$

- $Q(K_i)$ is the mid-quote for a strike of K_i, K_0 is the first strike below the forward index level
- VIX appears to have information about future realized volatility that is not in other backward looking measures (GARCH/RV)
- Computes area under curves defined by OOM options

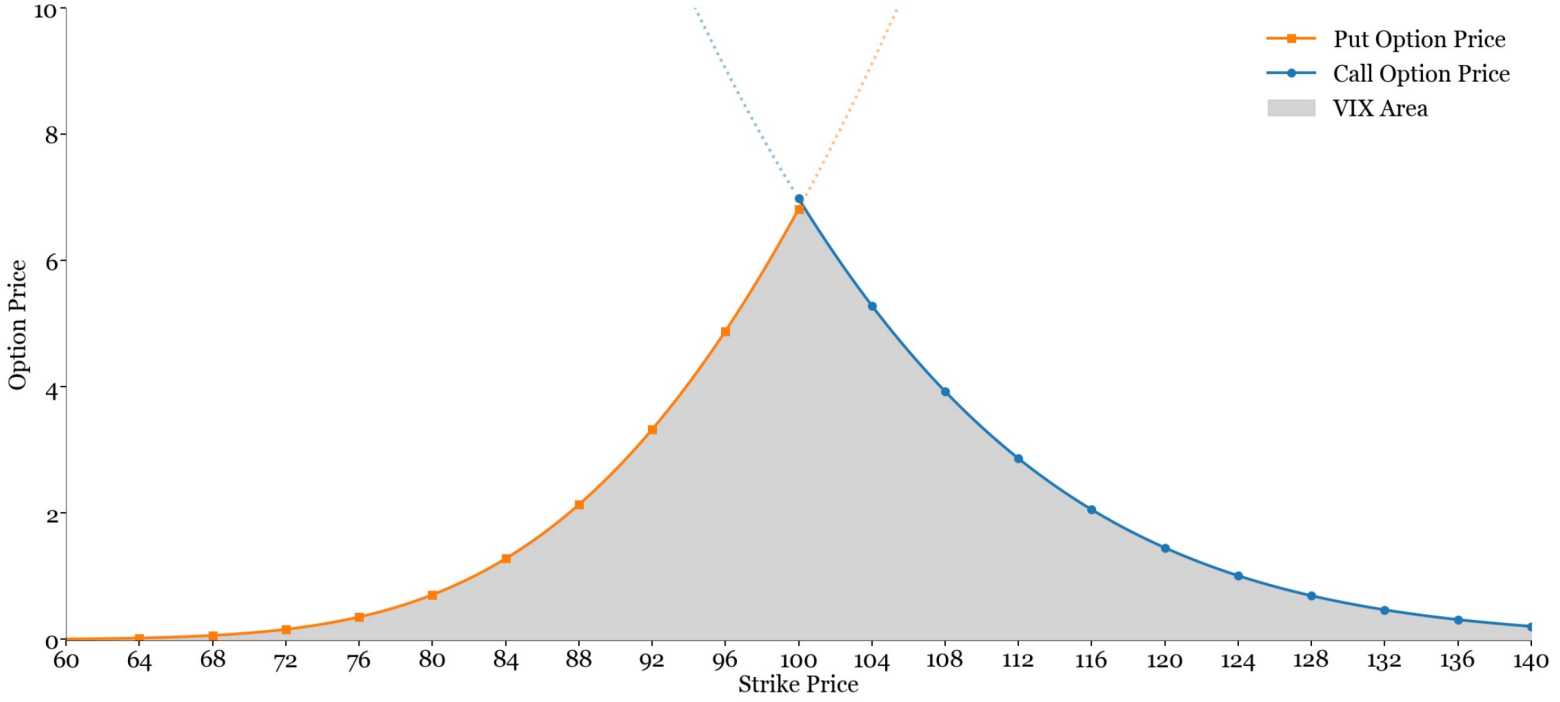
20% Annualized Volatility

In [4]:
```
plot_20()
```



60% Annualized Volatility

In [6]:
```
plot_60()
```



Value-at-Risk

- Defining VaR
- Changed in the future return distribution and VaR
- Volatility-based VaR models
- CaViaR
- Weighted Historical Simulation
- VaR model evaluation

Percentage Value-at-Risk

Defined

The α percentage Value-at-Risk (%VaR) of a portfolio is defined as the largest return such that the probability that the return on the portfolio over some period of time is less than -%VaR is α

$$\Pr(r < -\%\mathrm{VaR}) = \alpha$$

where r is the percentage return on the portfolio.

- I will use VaR interchangeably with %VaR
- Usually interested in *conditional* VaR

$$\Pr(r_{t+1} < -\%\mathrm{VaR}_{t+1} | \mathcal{F}_t) = \alpha$$

The Effect of Changes in Mean

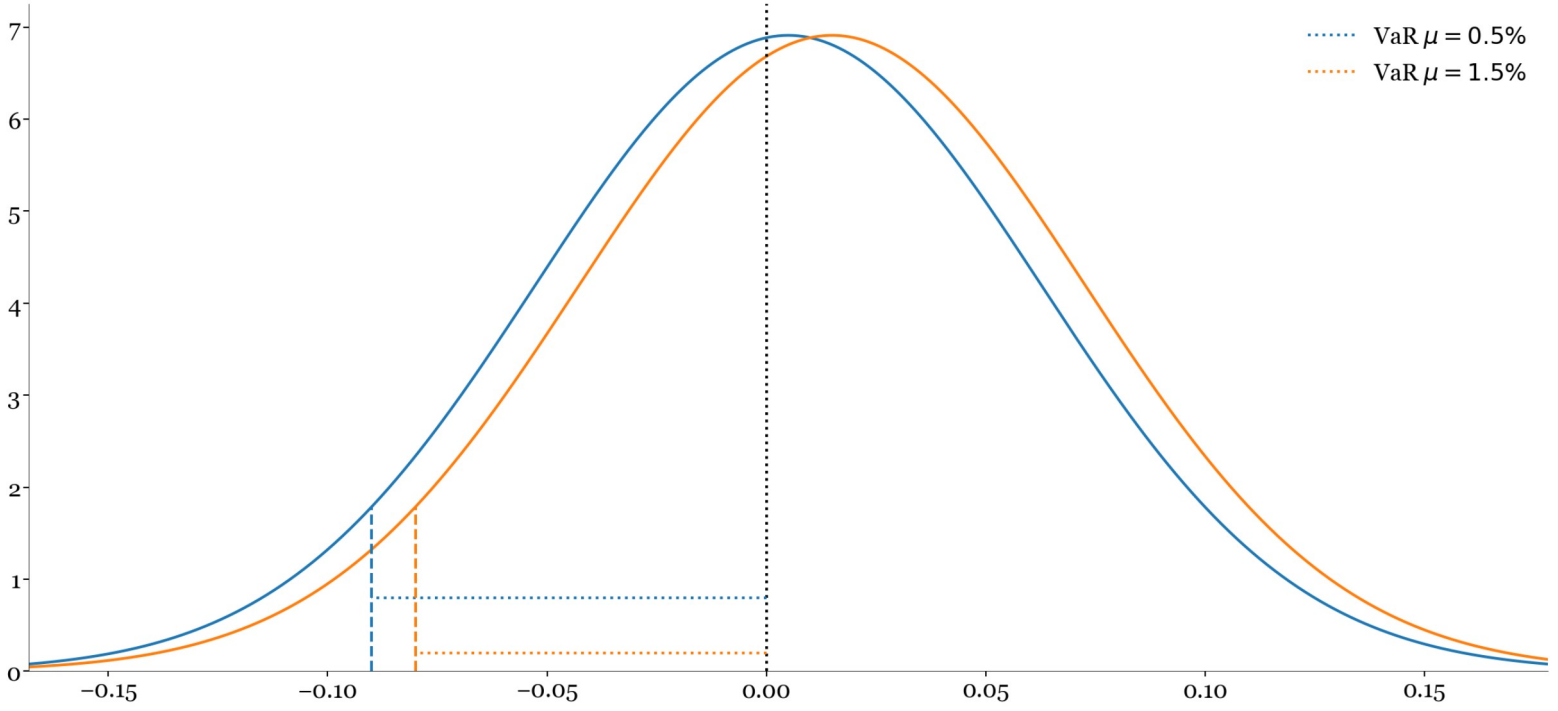
- VaR has an inverse relationship to the mean of a portfolio
- The mean always enters additively

$$\text{VAR} = -\mu + \text{other terms}$$

- Higher mean reduces VaR

Mean Shifts

In [9]:
```
plot_mean_shift()
```



The Effect of Changes in Standard Deviation

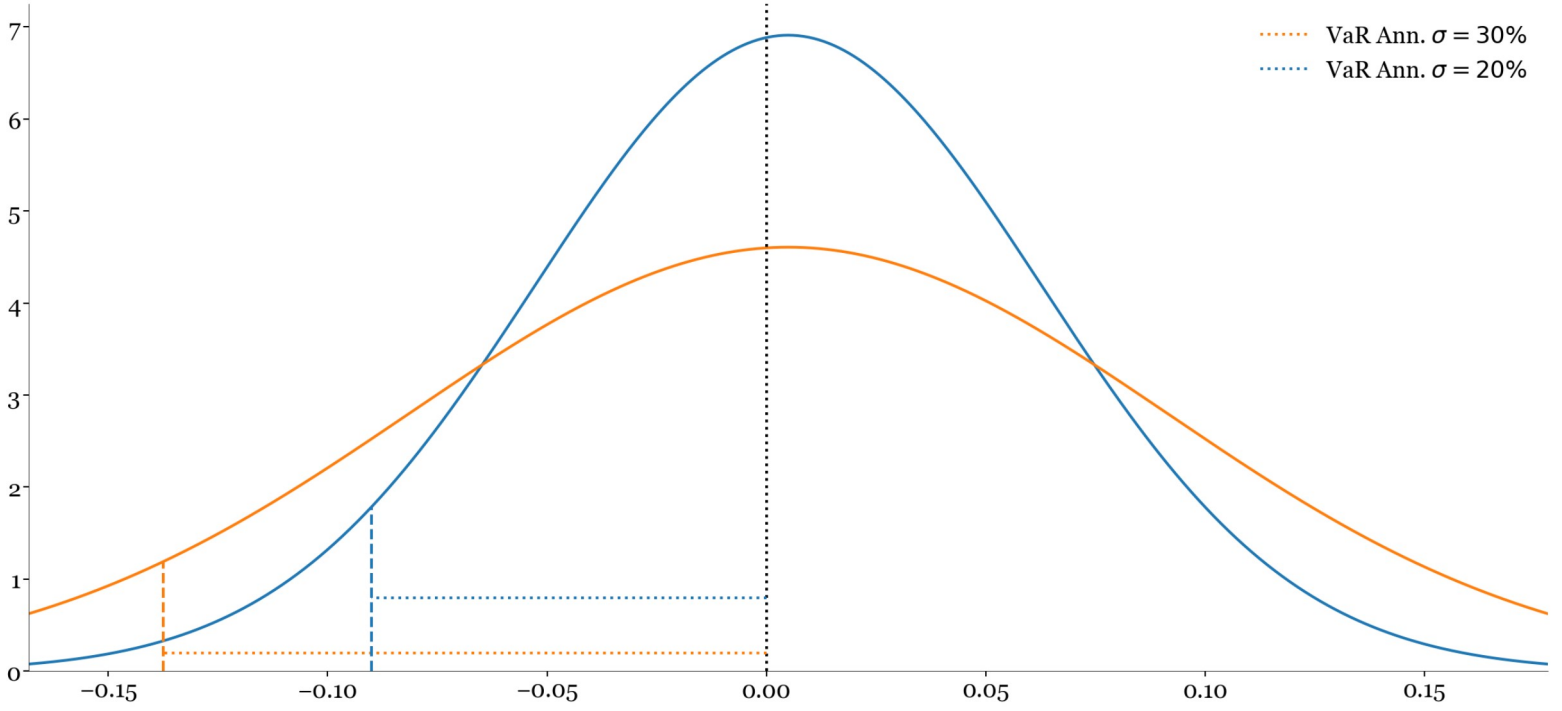
- VaR has an linear relationship to the standard deviation of a portfolio
- The standard deviation always enters multiplicatively

$$\text{VAR}_{\alpha} = -\mu - \sigma F^{-1}(\alpha)$$

- $F^{-1}(\alpha)$ is the *quantile* function
  - For relevant α (1%, 5%), $F^{-1}(\alpha)$ is *negative*
- Higher standard deviation increases VaR

Standard Deviation Changes

In [10]:
```
plot_std_change()
```



Connection to Quantiles

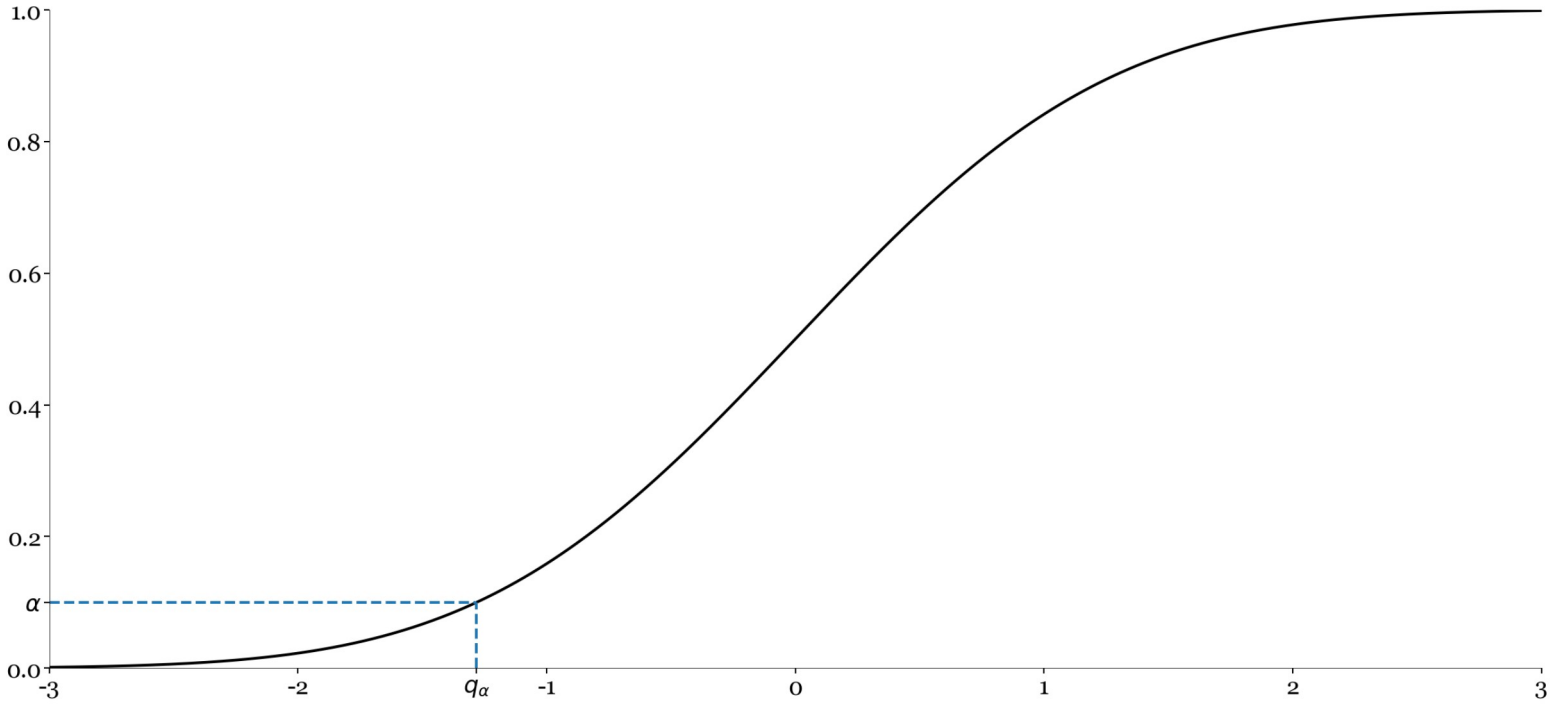
- The VaR crucially depends on the *quantile* of the future return distribution

$$VaR = -q_\alpha$$

- In principle only need to model q_α

Connection to Quantiles

In [12]:
```
quantile_plot()
```



RiskMetrics

- Exponentially Weighted Moving Average (EWMA) Variance:

$$\sigma_{t+1}^2 = (1-\lambda) r_t^2 + \lambda \sigma_t^2$$

$$\text{VaR}_{t+1} = -\sigma_{t+1} \Phi^{-1}(\alpha)$$

- $\Phi^{-1}(\cdot)$ is the inverse normal cdf
- λ is a known, fixed value
    - .94 (daily data)
    - .97 (weekly)
    - .99 (monthly)

RiskMetrics Variance Forecasts

In [14]:
```
fcast_variance.head()
```

Out[14]:

Date	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10
2000-06-29	1.458982	1.458982	1.458982	1.458982	1.458982	1.458982	1.458982	1.458982	1.458982	1.458982
2000-06-30	1.414438	1.414438	1.414438	1.414438	1.414438	1.414438	1.414438	1.414438	1.414438	1.414438
2000-07-03	1.392867	1.392867	1.392867	1.392867	1.392867	1.392867	1.392867	1.392867	1.392867	1.392867
2000-07-05	1.460259	1.460259	1.460259	1.460259	1.460259	1.460259	1.460259	1.460259	1.460259	1.460259
2000-07-06	1.403911	1.403911	1.403911	1.403911	1.403911	1.403911	1.403911	1.403911	1.403911	1.403911

RiskMetrics Value-at-Risk

- VaR scales with $\sqrt{h}$ rule
- Driven by assumption of constant variance and no serial correlation

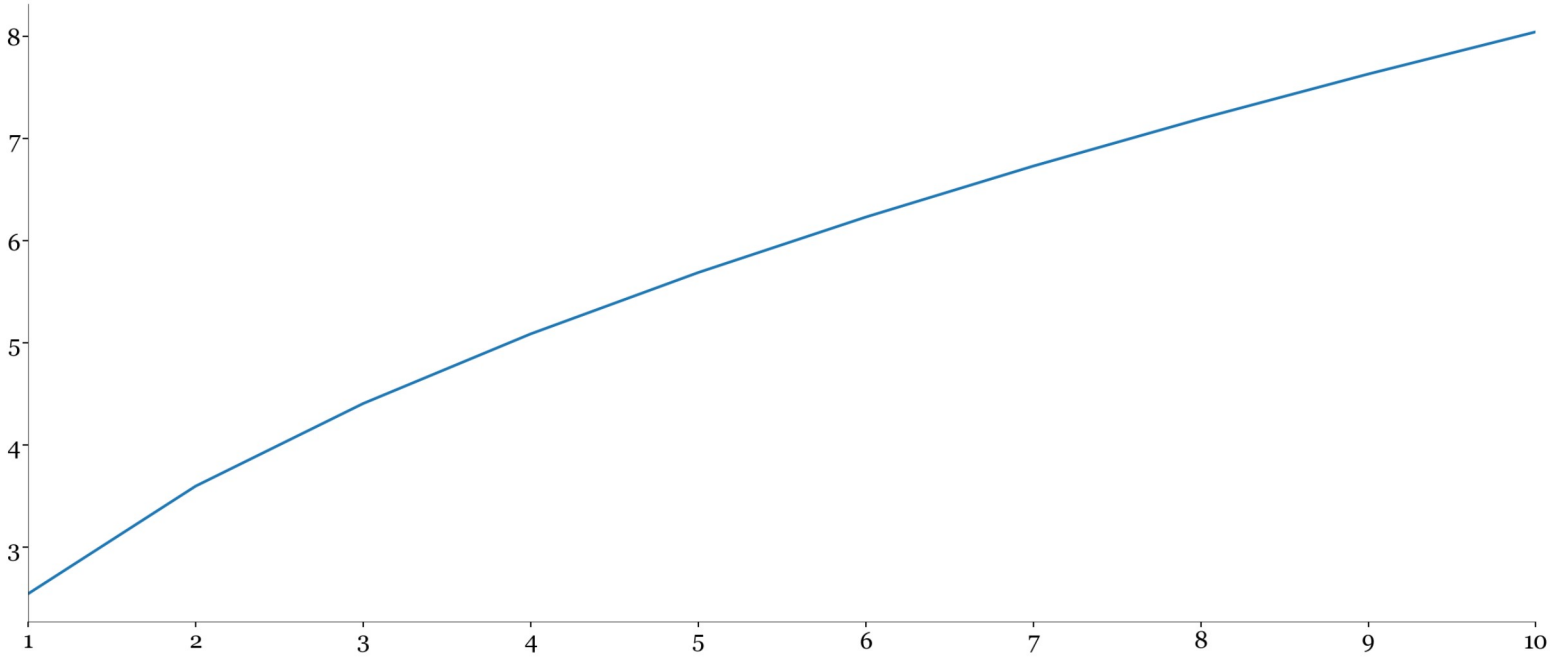
In [16]:
```
value_at_risk.head()
```

Out[16]:

Date	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10
2000-06-29	2.809957	3.973880	4.866989	5.619914	6.283255	6.882961	7.434448	7.947759	8.429871	8.885865
2000-06-30	2.766729	3.912746	4.792115	5.533458	6.186594	6.777075	7.320077	7.825492	8.300187	8.749166
2000-07-03	2.745551	3.882795	4.755433	5.491101	6.139238	6.725198	7.264044	7.765590	8.236652	8.682193
2000-07-05	2.811187	3.975618	4.869118	5.622373	6.286004	6.885973	7.437701	7.951236	8.433560	8.889752
2000-07-06	2.756414	3.898158	4.774248	5.512827	6.163528	6.751807	7.292785	7.796315	8.269241	8.716545

Risk Metrics Value-at-Risk

In [18]:
```
plot(rm_final)
```



Parameteric GARCH

$$r_{t+1} = \mu + \epsilon_{t+1}$$

$$\sigma^2_{t+1} = \omega + \gamma\epsilon^2_t + \beta\sigma^2_t$$

$$\epsilon_{t+1} = \sigma_{t+1} e_{t+1}$$

$$e_{t+1} \overset{iid}{\sim} F(0,1)$$

- Value-at-Risk

$$\mathrm{VaR}_{t+1} = -\hat{\mu} - \hat{\sigma}_{t+1} F_\alpha^{-1}$$

Skew-t GJR-GARCH

In [19]:

```
gjr = arch_model(sp500, o=1, power=2.0, dist="skewt").fit(disp="off")
summary(gjr, [1, 2])
```

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.0390	7.739e-03	5.041	4.632e-07	[2.384e-02,5.418e-02]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0161	2.717e-03	5.909	3.446e-09	[1.073e-02,2.138e-02]
alpha[1]	0.0209	4.811e-03	4.342	1.410e-05	[1.146e-02,3.032e-02]
gamma[1]	0.1331	1.536e-02	8.666	4.459e-18	[0.103, 0.163]
beta[1]	0.8999	8.634e-03	104.231	0.000	[0.883, 0.917]

Variance forecast is mean reverting

In [21]:
```
fcast_variance.head()
```

Out[21]:

Date	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10
2000-06-29	1.136292	1.138025	1.139736	1.141426	1.143094	1.144741	1.146367	1.147973	1.149558	1.151124
2000-06-30	1.052276	1.055068	1.057825	1.060547	1.063234	1.065888	1.068508	1.071095	1.073650	1.076172
2000-07-03	0.983440	0.987100	0.990713	0.994281	0.997804	1.001282	1.004717	1.008108	1.011457	1.014763
2000-07-05	1.307899	1.307469	1.307044	1.306624	1.306210	1.305801	1.305397	1.304998	1.304604	1.304216
2000-07-06	1.202832	1.203726	1.204609	1.205481	1.206341	1.207191	1.208030	1.208859	1.209677	1.210485

Skew t GJR-GARCH Value-at-Risk

- Scaling is not $\sqrt{h}$
- Simple method ignores effect of changing variance

In [23]:
```
print(f"Skew t alpha={alpha:0.02f} Quantile: {q:0.3f}")
```

```
Skew t alpha=0.01 Quantile: -2.655
```

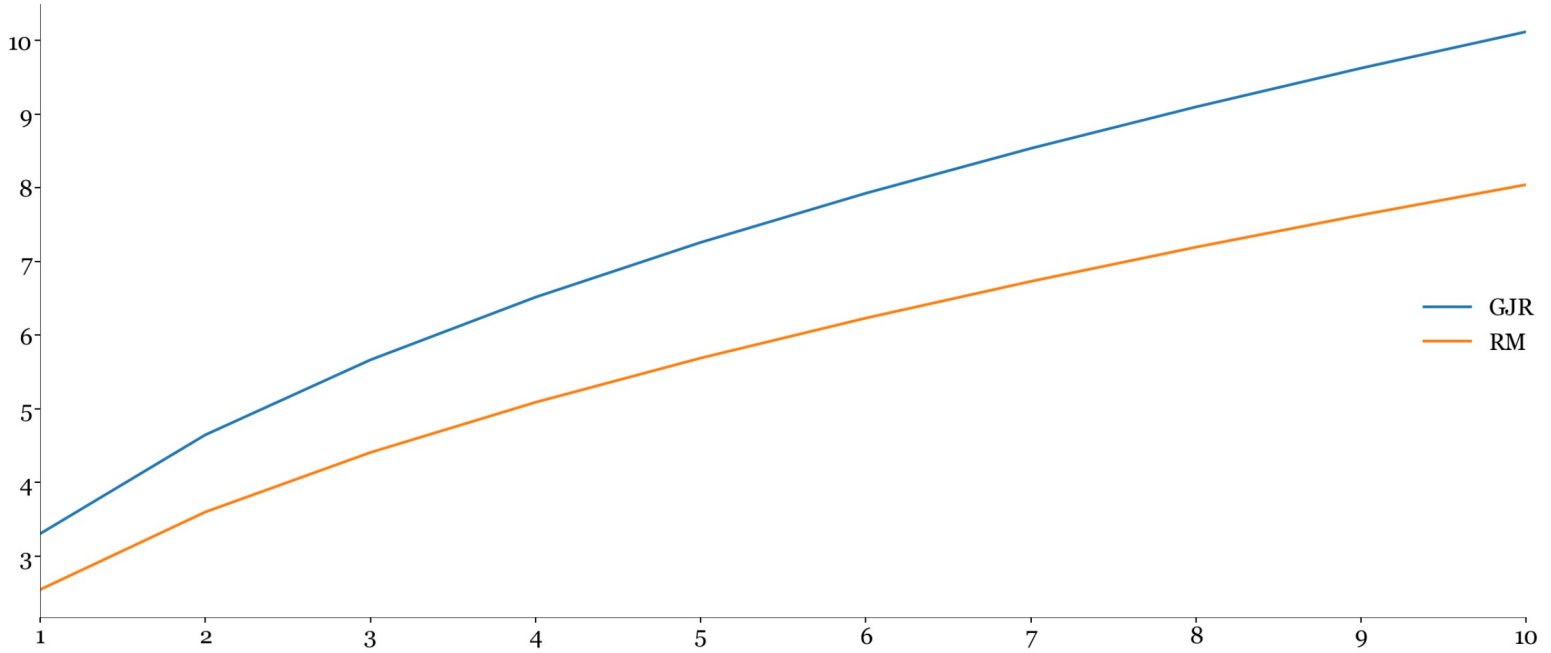
In [24]:
```
gjr_value_at_risk.head()
```

Out[24]:

Date	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10
2000-06-29	2.790671	3.925279	4.787841	5.509737	6.141830	6.710195	7.230313	7.712268	8.163059	8.587780
2000-06-30	2.684051	3.775523	4.605676	5.300817	5.909830	6.457766	6.959493	7.424692	7.860080	8.270544
2000-07-03	2.593478	3.648342	4.451013	5.123486	5.712962	6.243619	6.729804	7.180854	7.603247	8.001693
2000-07-05	2.996835	4.214960	5.140346	5.914163	6.591099	7.199202	7.755137	8.269762	8.750621	9.203207
2000-07-06	2.872344	4.040021	4.927448	5.669885	6.319711	6.903783	7.438052	7.932910	8.395570	8.831286

Skew t GJR-GARCH VaR

In [26]:
```
plot(df, loc=5)
```



Filtered Historical Simulation

$$r_{t+1} = \mu + \epsilon_{t+1}$$

$$\sigma^2_{t+1} = \omega + \gamma\epsilon^2_t + \beta\sigma^2_t$$

$$\epsilon_{t+1} = \sigma_{t+1} e_{t+1}$$

$$e_{t+1} \overset{iid}{\sim} G(0,1)$$

- G unknown
- Use empirical CDF to directly estimate quantile

Empirical CDFs

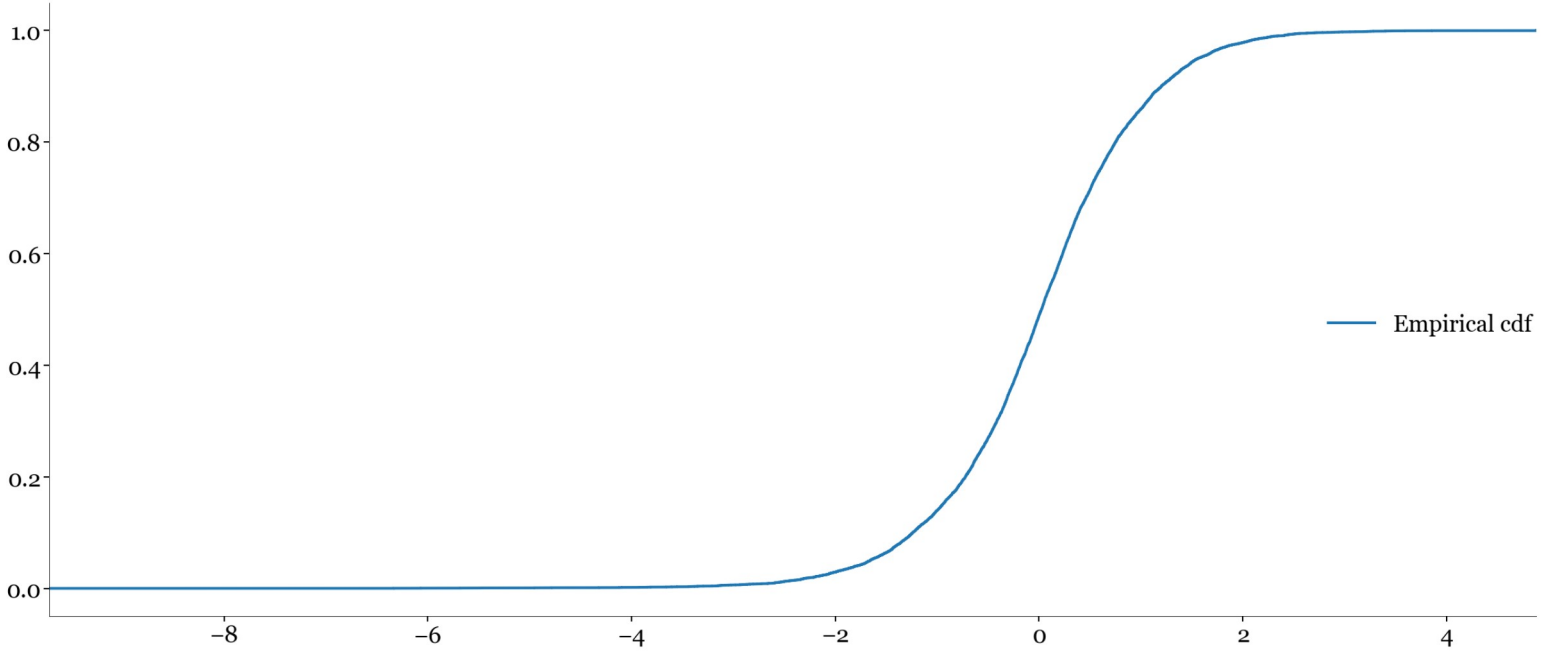
- Empirical CDFs are step functions
- Formally defined as

$$\hat{G}(x) = \frac{1}{n} \sum_{i=1}^{n} I_{[x<x_i]}$$

- In practice trivial to invert using the sorted values x_i

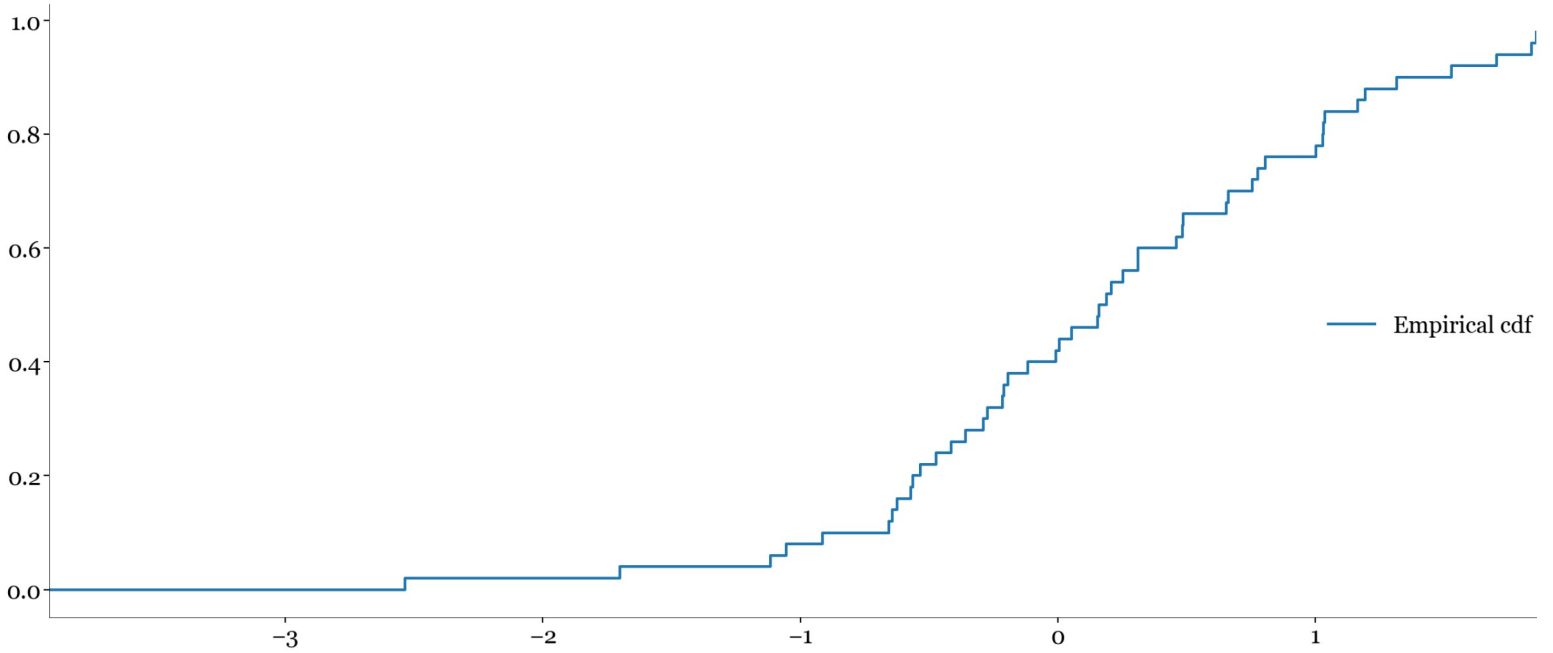
The Empirical cdf of e

In [29]:
```
e = gjr_normal.std_resid
cdf_plot(e)
```



The Empirical cdf of e

In [30]:
```
cdf_plot(e.iloc[-50:])
```



FHS VaR

In [32]:
```
print(f"Empirical cdf alpha={alpha} Quantile: {q:0.3f}")
```

```
Empirical cdf alpha=0.01 Quantile: -2.590
```

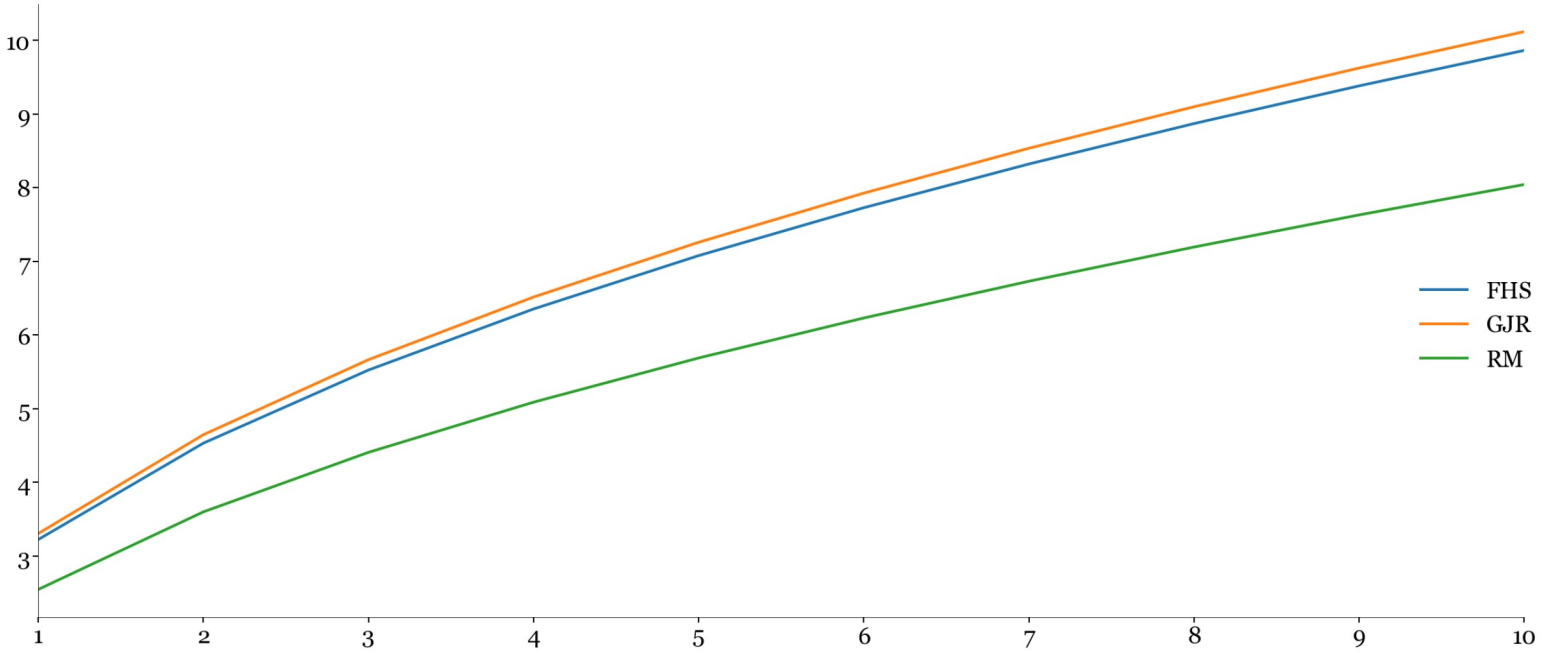
In [33]:
```
gjr_fhs_value_at_risk.head()
```

Out[33]:

Date	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10
2000-06-29	2.721996	3.828120	4.668801	5.372230	5.988035	6.541659	7.048208	7.517519	7.956422	8.369889
2000-06-30	2.617963	3.681998	4.491057	5.168380	5.761666	6.295357	6.783960	7.236922	7.660796	8.060352
2000-07-03	2.529588	3.557904	4.340147	4.995353	5.569575	6.086407	6.559845	6.999002	7.410197	7.798026
2000-07-05	2.923156	4.110770	5.012751	5.766841	6.426401	7.018798	7.560294	8.061483	8.529725	8.970379
2000-07-06	2.801686	3.940077	4.805020	5.528491	6.161599	6.730549	7.250905	7.732806	8.183291	8.607485

Comparing VaR Models

In [35]:
```
plot(df, loc=5)
```



CaViaR

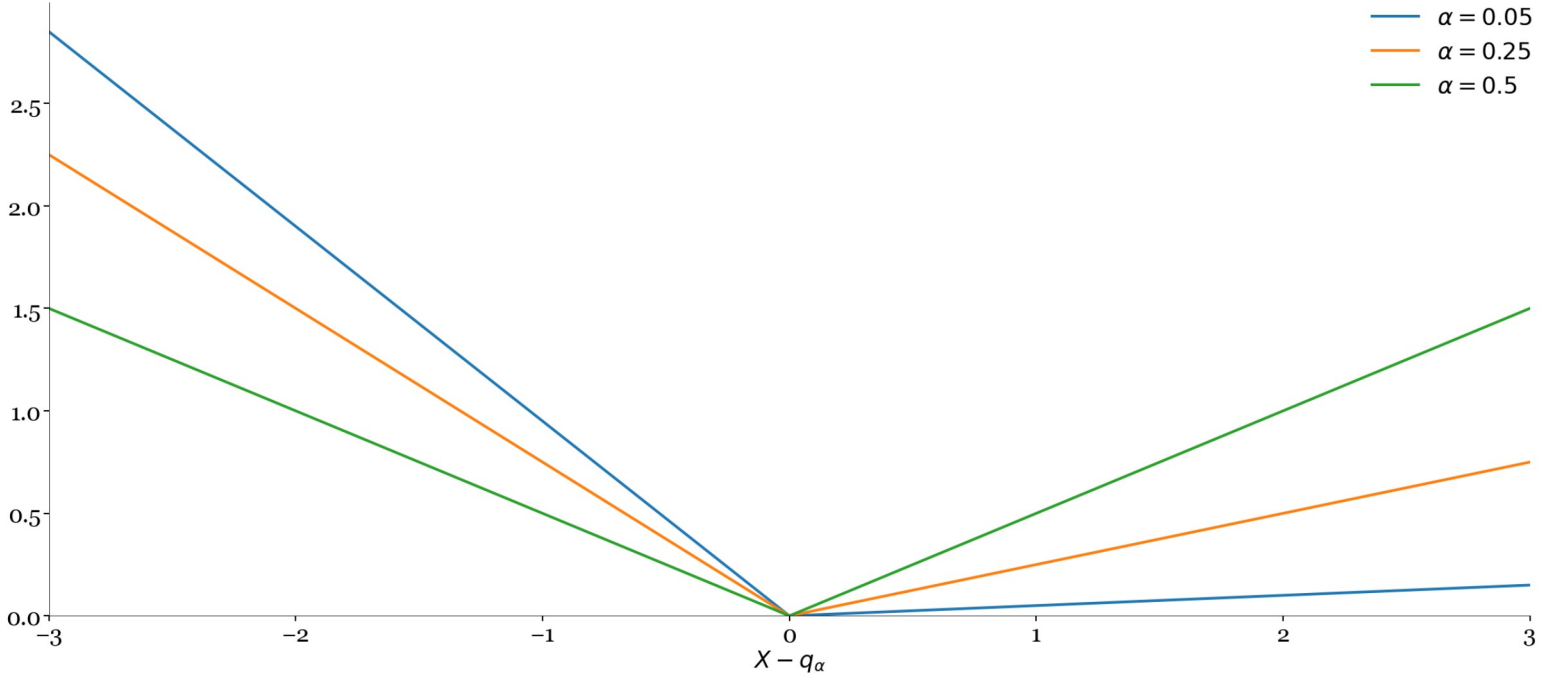
- Conditional quantile regression
- Uses ARCH-like dynamics
- Minimizes the *tick* loss function

$$\arg\min_{\theta} \; T^{-1} \sum_{t=1}^{T} \alpha(r_t - q_t)(1 - I_{[r_t < q_t]}) + (1-\alpha)(q_t - r_t) I_{[r_t < q_t]}$$

- q_t is the conditional α quantile of r_t

Tick Loss Function

In [37]:
```
tick_loss()
```



CaViar: Symmetric

- Quantiles evolove according to

$$q_{t+1} = \omega + \gamma \text{HIT}_t + \beta q_t$$

- A HIT is defined

$$\text{HIT}_t = I_{[r_t < q_t]} - \alpha$$

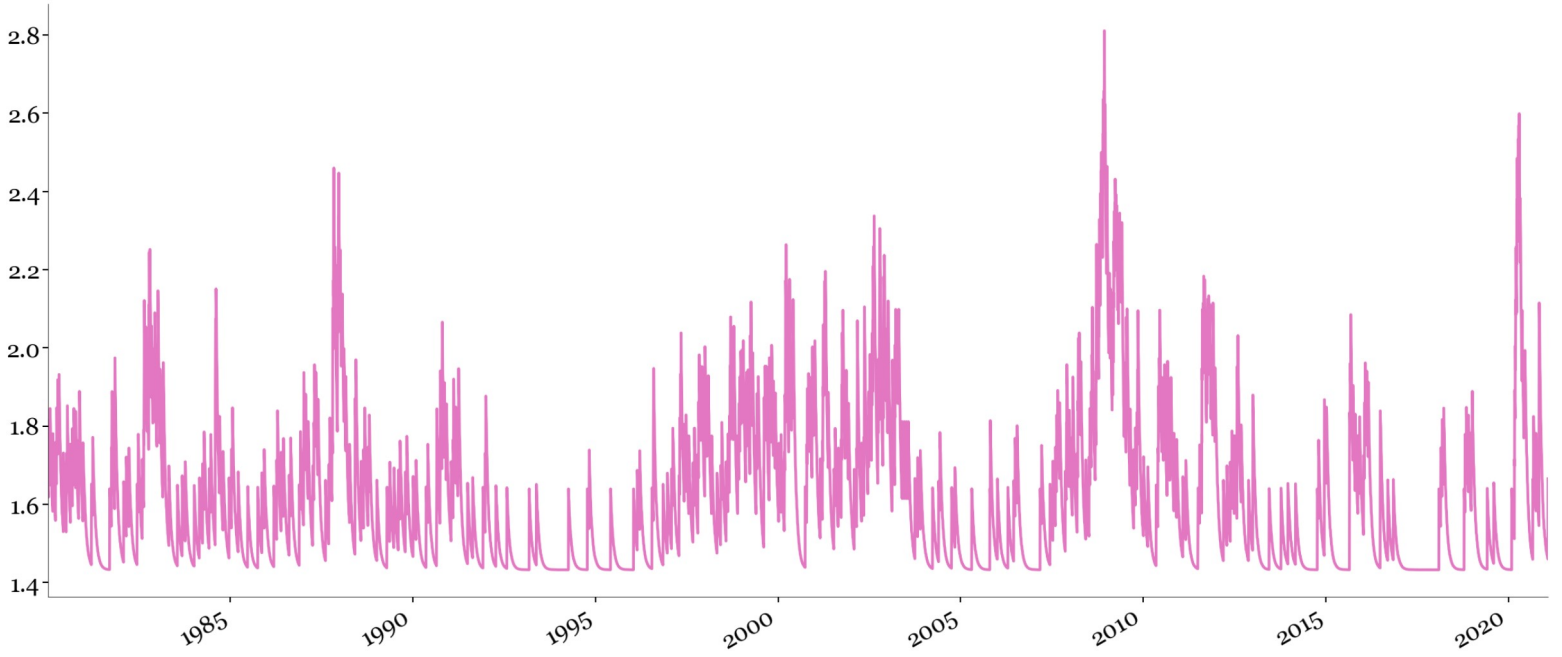
- Mean-zero, white-noise shock under correct specification

In [39]:
```
opt = caviar.caviar_hit(rets, disp=False)
pd.Series(opt.x, index=["omega", "alpha", "beta"])
```

Out[39]:
```
omega    0.078519
alpha    0.206854
beta     0.945180
dtype: float64
```

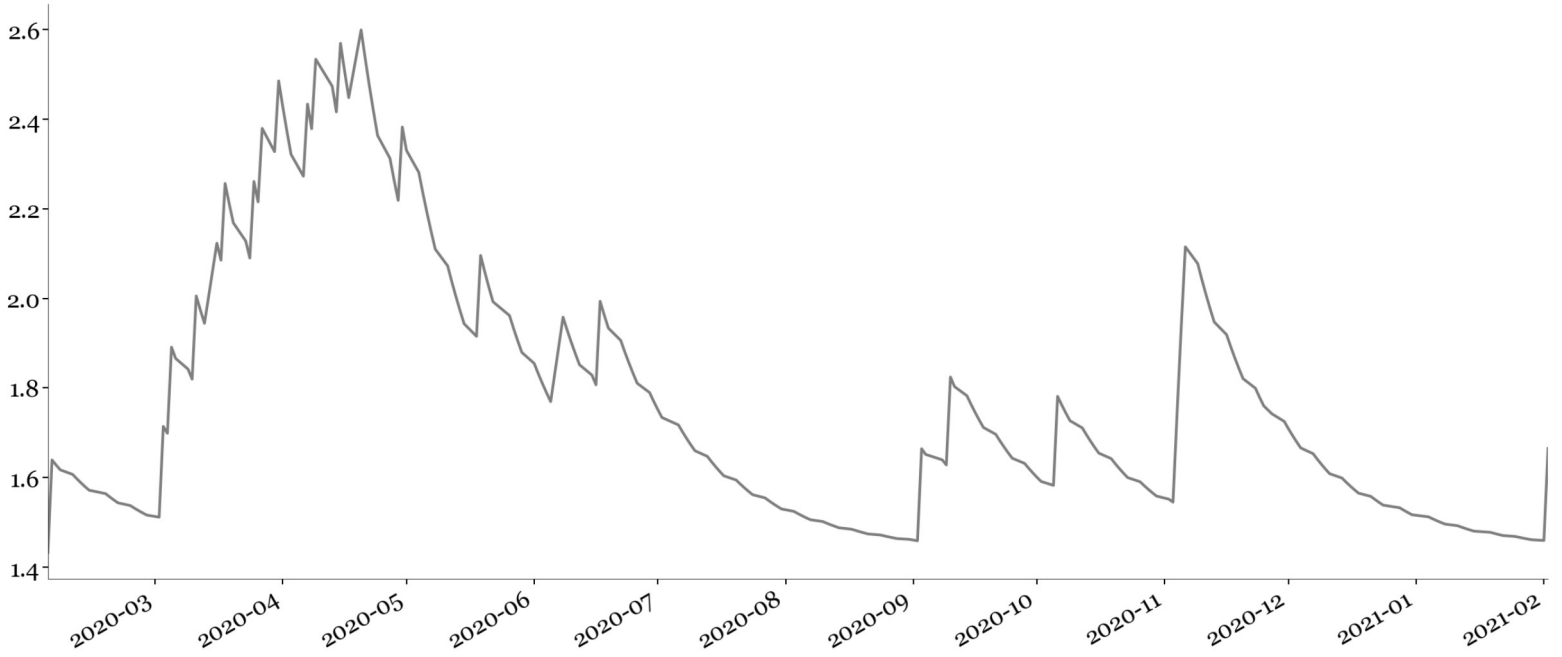
Conditional Quantiles

In [41]:
```
plot(s)
```



Conditional Quantiles (past year)

In [42]:
```
plot(s.iloc[-252:])
```



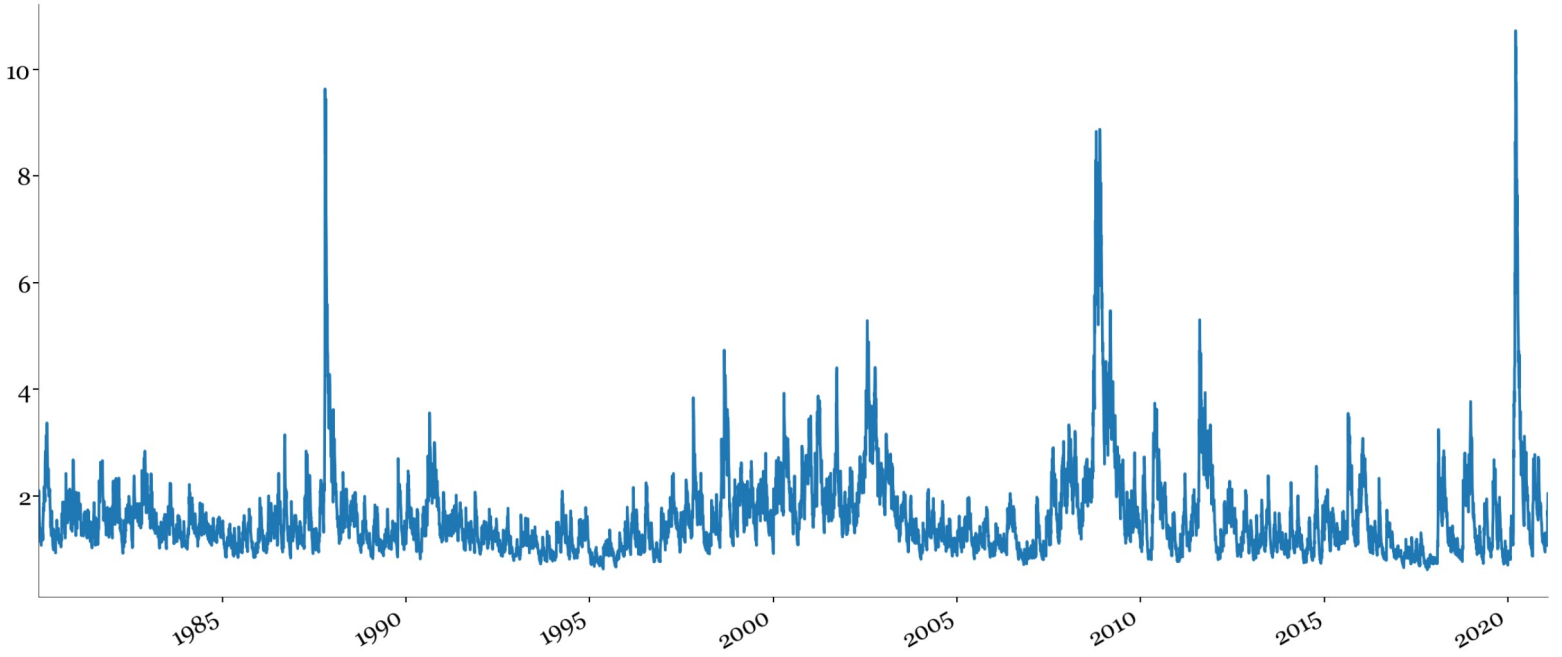
CaViaR: Asymmetric absolute value

In [44]:
```
pd.Series(opt.x, index=["omega", "alpha", "gamma", "beta"])
```

Out[44]:
```
omega     0.046263
alpha     0.036145
gamma     0.241215
beta      0.900651
dtype: float64
```

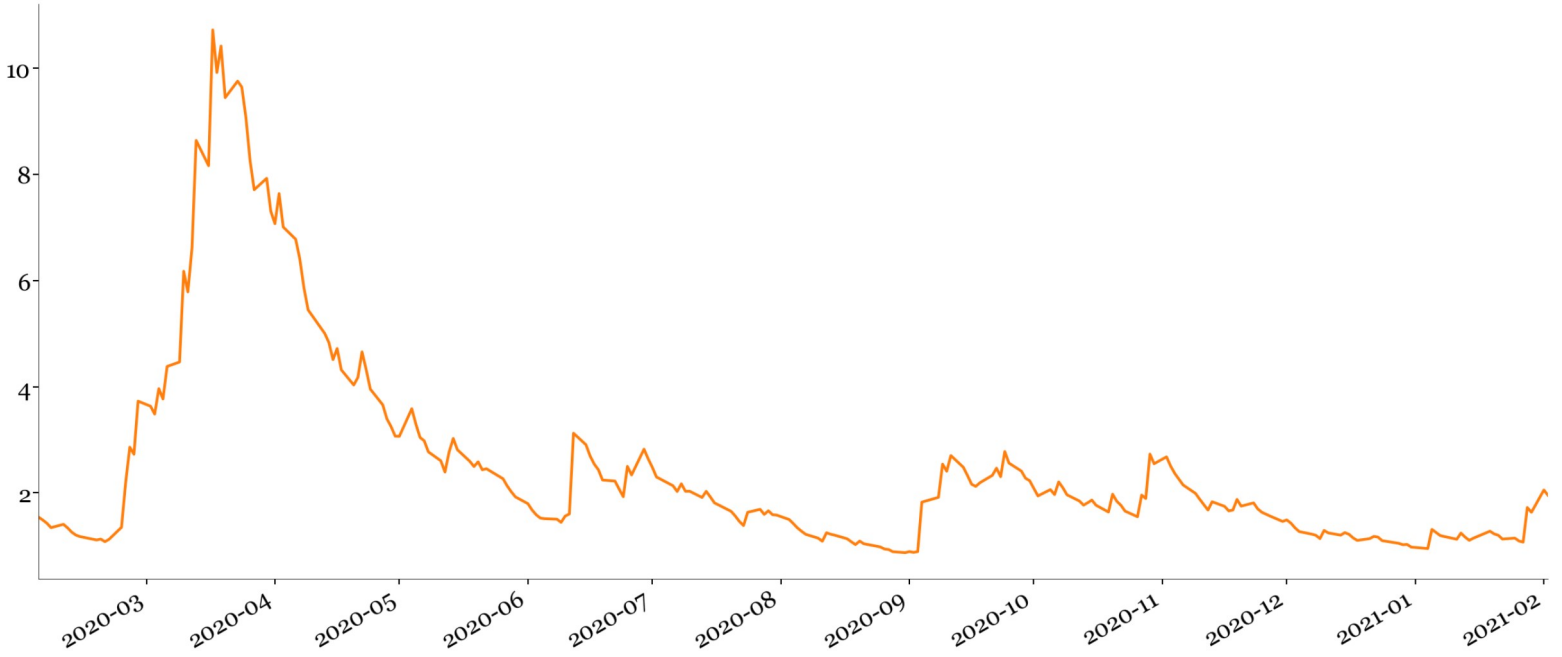
CaViaR: Asymmetric absolute value quantiles

In [45]:
```
plot(asym_caviar_q)
```



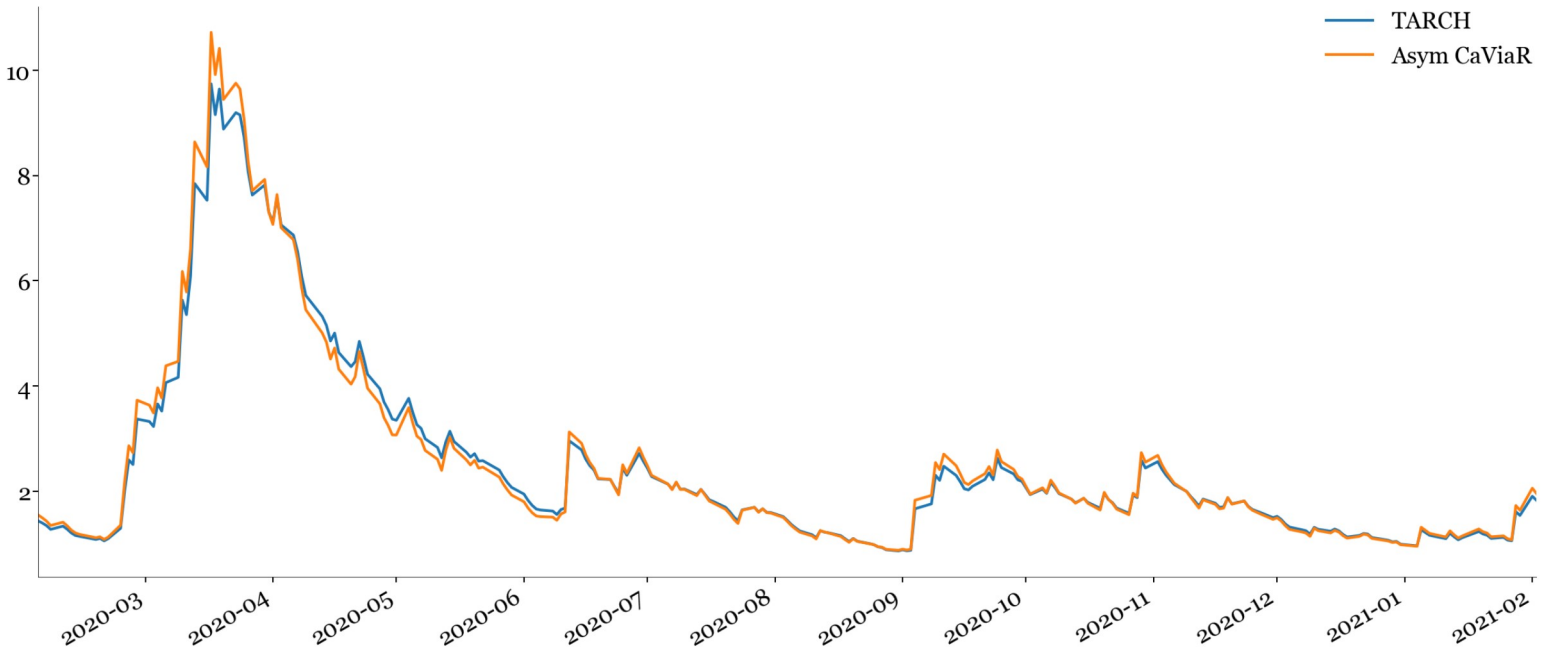
CaViaR: Asymmetric absolute value quantiles (last year)

In [46]:
```
plot(asym_caviar_q.iloc[-252:])
```



Comparing Asym Absolute Val CaViaR and Skew t TARCH

In [48]:
```
plot(df.iloc[-252:])
```



Weighted Historical Simulation

$$w_i = \lambda^{t-i}(1-\lambda) / \left(1-\lambda^t\right) \ , \ i = 1,2,\ldots,t$$

- Weighted Empirical cdf

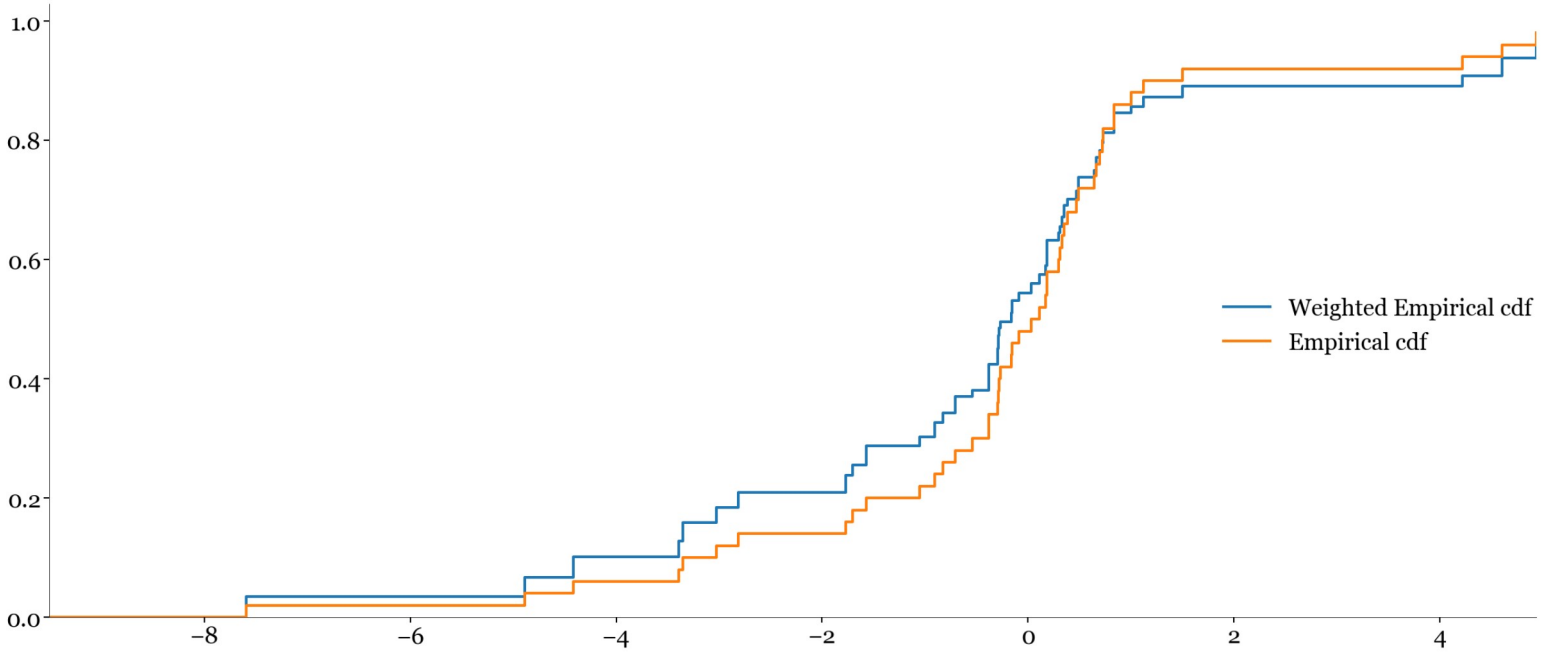
$$\hat{G}_t(r) = \sum_{i=1}^{t} w_i I_{[r_i<r]}$$

- Conditional VaR is solution to

$$\mathrm{VaR}_{t+1} = -\min_r \hat{G}(r) \geq \alpha$$

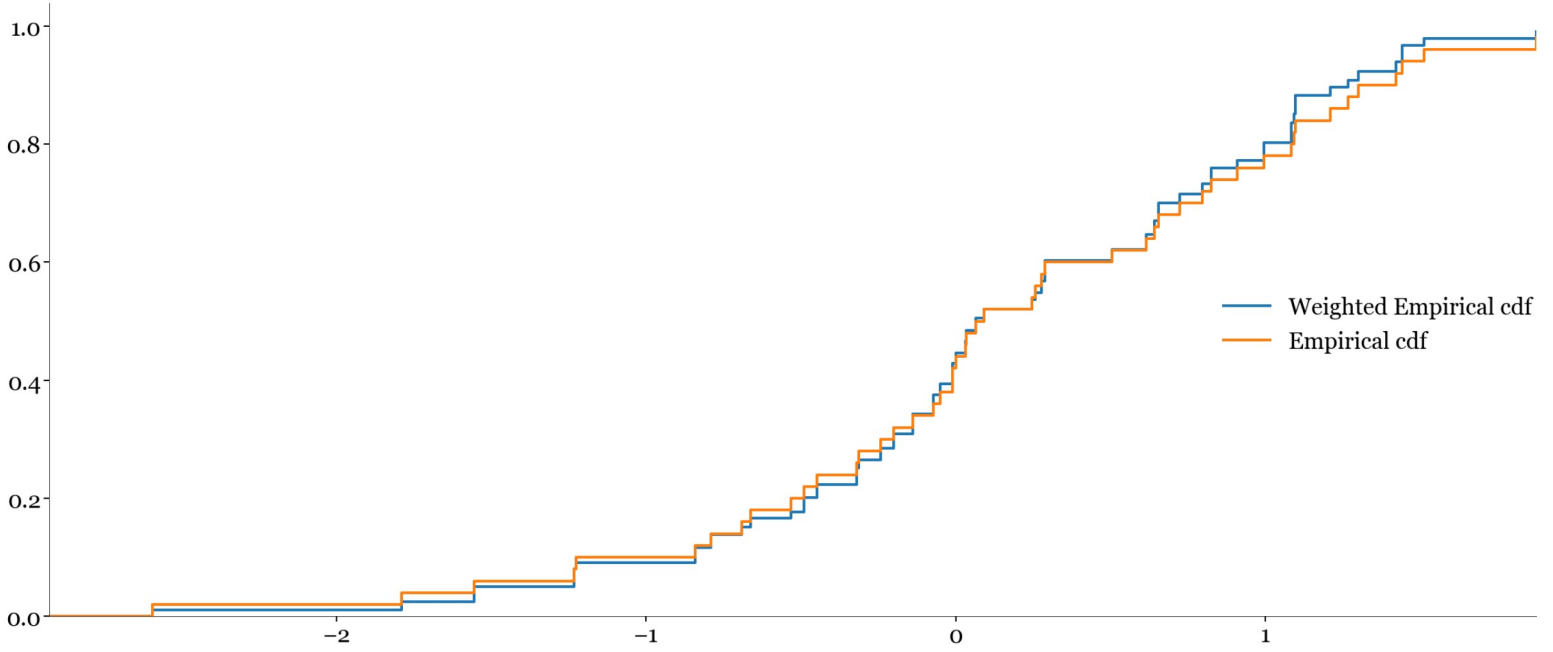
Weighted ECDF at start of COVID Pandemic

In [51]:
```
plot_wecdf(sp500.iloc[-275:-225], 0.975, True)
```



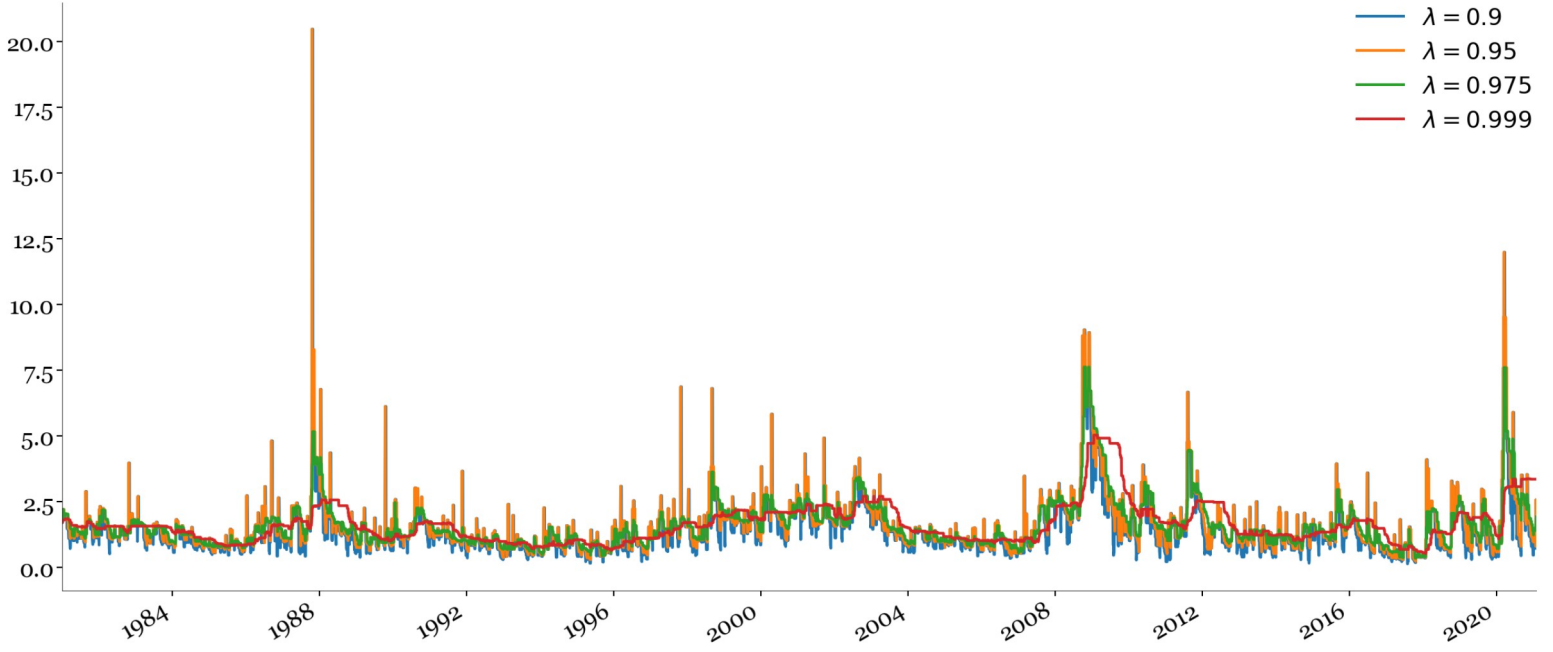
Weighted ECDF before start of COVID Pandemic

In [52]:
```
plot_wecdf(sp500.iloc[-375:-325], 0.975, True)
```



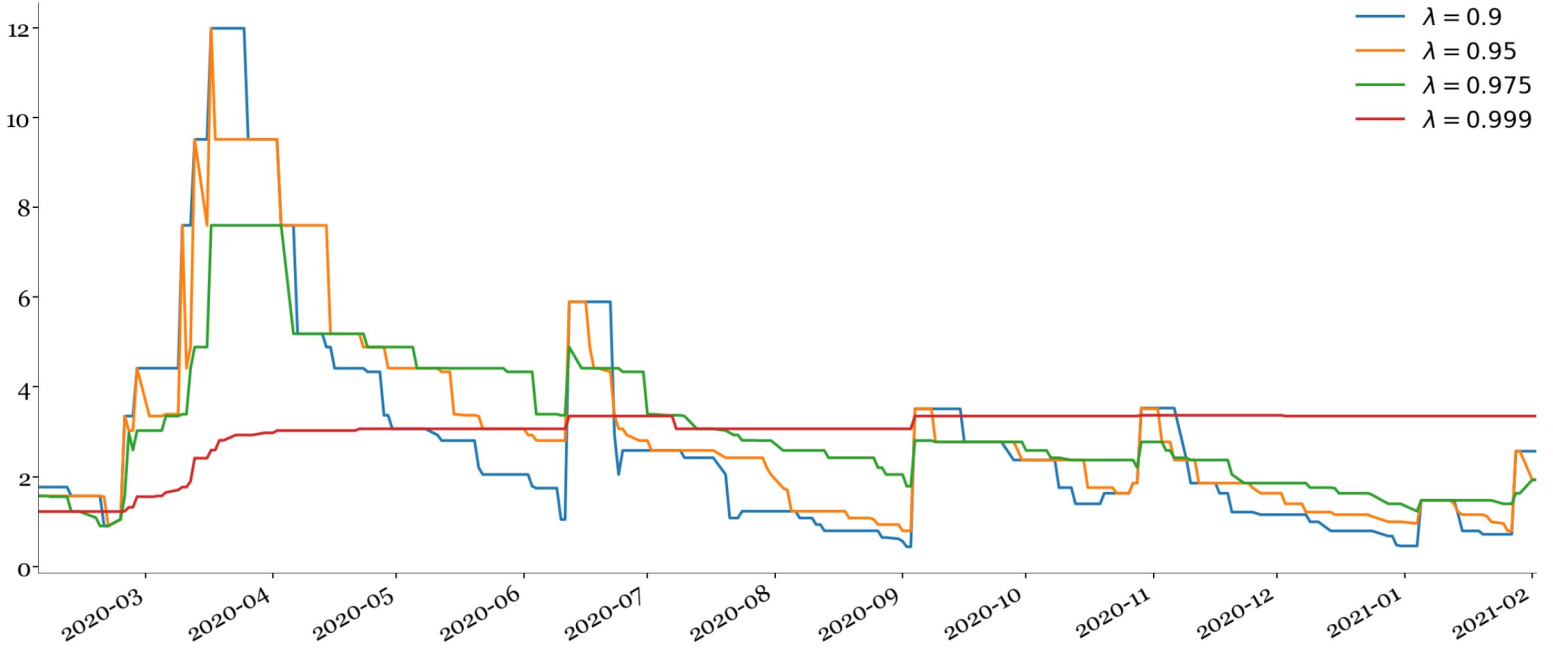
Weighted HS

In [53]:
```
plot(df)
```



Weighted HS (last year)

In [54]:
```
plot(df.iloc[-252:])
```



Evaluation of Value-at-Risk Models

- Generalized Mincer-Zarnowitz

$$\mathrm{HIT}_{t \mid h} = \gamma_0 + \gamma_1 VaR_{t+h|t} + \gamma_2 \mathrm{HIT}_t + \gamma_3 \mathrm{HIT}_{t-1} + \ldots + \gamma_K \mathrm{HIT}_{t-K \mid 1} + \eta_t$$

- Bernoulli

$$I_{[r_t < q_t]} \sim \mathrm{Bernoulli}(\alpha)$$

- Christoffersen's Conditional Bernoulli Test

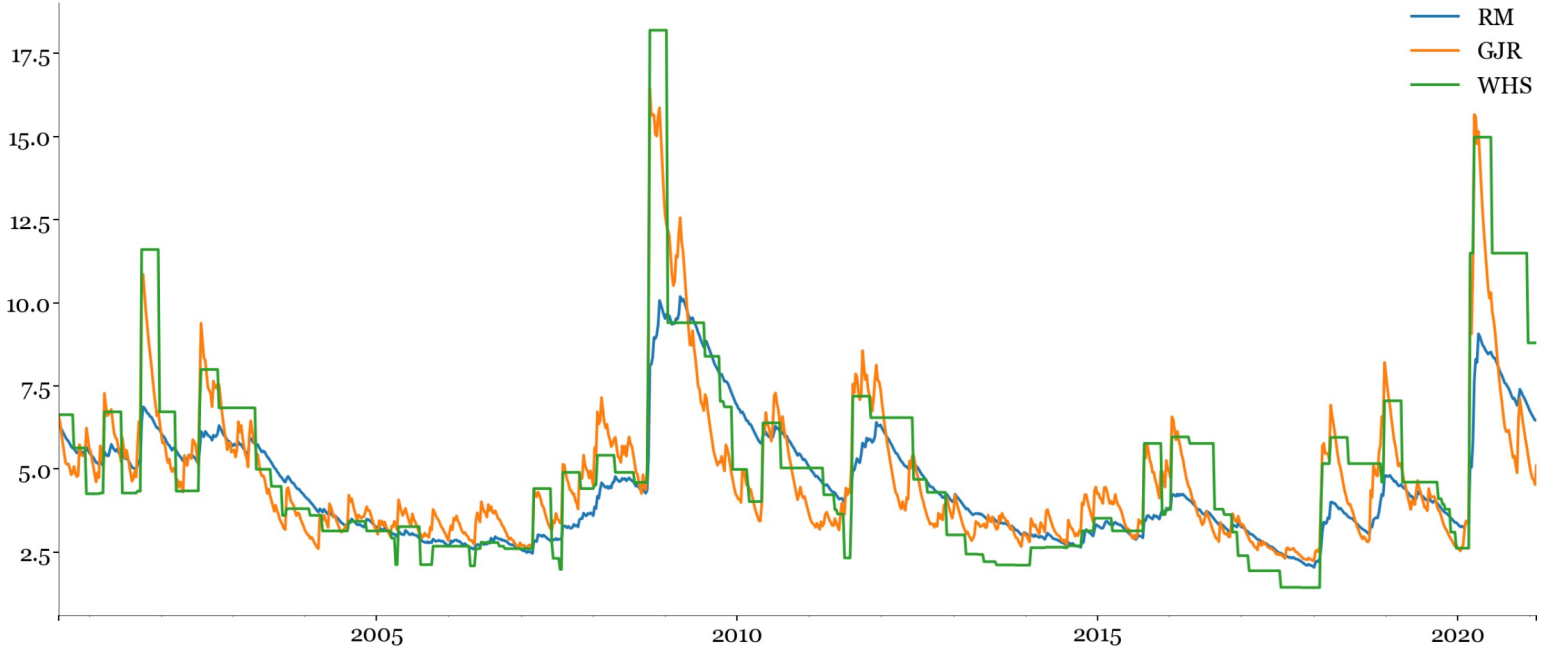
$$I_{[r_t < q_t]} \big| \, r_{t-1}, q_{t-1} \sim \mathrm{Bernoulli}(\alpha)$$

- Logit/Probit improvement of GMZ

$$I_{[r_t < q_t]} \big| \, \mathcal{F}_{t-1} \sim \mathrm{Bernoulli}(\alpha)$$

Weekly VaRs for S&P 500

In [56]:
```
plot(weekly_vars)
```



Evaluation: GMZ Regression

$$\mathrm{HIT}_{t+h} = \gamma_0 + \gamma_1 VaR_{t+h|t} + \gamma_2 \mathrm{HIT}_t + \gamma_3 \mathrm{HIT}_{t-1} + \ldots + \gamma_K \mathrm{HIT}_{t-K+1} + \eta_t$$

GMZ for Skew t GJR

In [59]:
```
gjr_res = gmz(hits, weekly_vars, "GJR")
summary(gjr_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0105	0.010	1.020	0.308	-0.010	0.031
VaR	-0.0014	0.002	-0.797	0.426	-0.005	0.002
HIT.L1	0.0718	0.054	1.339	0.180	-0.033	0.177
HIT.L2	0.0009	0.034	0.027	0.978	-0.066	0.067
HIT.L3	0.0079	0.031	0.255	0.799	-0.053	0.069
HIT.L4	-0.0259	0.007	-3.568	0.000	-0.040	-0.012
HIT.L5	-0.0231	0.006	-4.015	0.000	-0.034	-0.012

In [60]:
```
joint_test(gjr_res)
```

Out[60]:

	stat	P-value
Joint Test	4.823941	0.000072

GMZ for RiskMetrics

In [61]:

```
rm_res = gmz(hits, weekly_vars, "RM")
summary(rm_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0457	0.014	3.156	0.002	0.017	0.074
VaR	-0.0083	0.002	-3.469	0.001	-0.013	-0.004
HIT.L1	0.0600	0.052	1.146	0.252	-0.043	0.163
HIT.L2	0.0851	0.056	1.511	0.131	-0.025	0.195
HIT.L3	0.0296	0.045	0.662	0.508	-0.058	0.117
HIT.L4	0.0897	0.056	1.604	0.109	-0.020	0.199
HIT.L5	-0.0265	0.027	-0.972	0.331	-0.080	0.027

In [62]:

```
joint_test(rm_res)
```

Out[62]:

	stat	P-value
Joint Test	3.544817	0.001762

GMZ for WHS

In [63]:
```
whs_res = gmz(hits, weekly_vars, "WHS")
summary(whs_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0323	0.011	2.892	0.004	0.010	0.054
VaR	-0.0049	0.001	-3.415	0.001	-0.008	-0.002
HIT.L1	0.0845	0.054	1.578	0.115	-0.020	0.189
HIT.L2	-0.0099	0.030	-0.329	0.742	-0.069	0.049
HIT.L3	0.0566	0.048	1.189	0.235	-0.037	0.150
HIT.L4	0.0239	0.040	0.592	0.554	-0.055	0.103
HIT.L5	-0.0297	0.008	-3.719	0.000	-0.045	-0.014

In [64]:
```
joint_test(whs_res)
```

Out[64]:

	stat	P-value
Joint Test	4.33738	0.000247

Evaluation: Unconditional Bernoulli Testing

- Likelihood of T exceedences

$$l(\alpha; \widetilde{\mathrm{HIT}}_t) = \sum_{t=1}^{T} \widetilde{\mathrm{HIT}}_t \ln \alpha + \left(1 - \widetilde{\mathrm{HIT}}_t\right) \ln 1 - \alpha$$

- $\widetilde{\mathrm{HIT}}$ is the indicator for a violation
- Easy to conduct a LR test

$$LR = 2(l(\hat{\alpha}; \widetilde{\mathrm{HIT}}) - l(\alpha_0; \widetilde{\mathrm{HIT}})) \sim \chi_1^2$$

In [66]:
```
pd.DataFrame(lrs).T
```

Out[66]:

	LR	P_value
RM	4.280032	0.038563
GJR	0.651270	0.419659
WHS	2.947361	0.086018

Conditional Bernoulli Test

Christoffersen's test

- Probability of a violation is independent of previous violation
- Leads to conditional Bernoulli distribution
- Has a close form expression
- Key inputs all depend on pairs $\mathbb{1}_{[r_t<q_t]}$ and $\mathbb{1}_{[r_{t-1}<q_{t-1}]}$

In [68]:
```
pd.DataFrame(cond_lrs).T
```

Out[68]:

	LR	P_value
RM	8.109426	0.017340
GJR	4.000991	0.135268
WHS	7.424546	0.024422

Evaluation: Using Logit to improve GMZ

- Same model as GMZ
- Exploit structure of exceedence to estimate using MLE
- Requires ensuring conditional probability is always $\in (0, 1)$
  - Logit uses logistic function $\Lambda(\cdot)$
  - Probit uses normal cdf $\Phi(\cdot)$

In [71]:
```
summary(logit_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.2243	0.416	-7.751	0.000	-4.040	-2.409
VaR	-0.0814	0.080	-1.024	0.306	-0.237	0.074
HIT.L1	1.4990	0.647	2.315	0.021	0.230	2.768
HIT.L2	0.1203	1.110	0.108	0.914	-2.056	2.296
HIT.L3	0.2558	0.991	0.258	0.796	-1.686	2.198

In [73]:
```
joint_logit_stat()
```

Out[73]:

	stat	p-value
Joint Test	9.267278	0.09886662583981386

Next Week

- Expected Shortfall
- Density Forecasting
- Coherent Risk Measures